PATENT APPLICATION FOR

VECTOR LOGIC TECHNIQUES FOR MULTILEVEL MINIMIZATION

****

INVENTORS:

Jonathan Westphal
7620 Valley Vista Road
Pocatello, ID 83201

****

VECTOR LOGIC TECHNIQUES FOR MULTILEVEL MINIMIZATION
Patent Application
Inventor: Jonathan Westphal

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to and incorporates by reference in their entirety, the following patent applications:

Provisional Application No. 60/408,938 filed September 6, 2002,

PCT Application US99/21955 filed September 22, 1999, and published as W000/17788 on March 30, 2000 by Jonathan Westphal, and

PCT Application US01/31217 filed October 5, 2001, and published as W002/29520 on April 11, 2002 by Jonathan Westphal.

FIELD OF THE INVENTION

This invention relates to the field of minimization for logical circuits, including but not limited to integrated circuits and optical logics.

DESCRIPTION OF RELATED ART

Minimization of logical circuits has been a long time goal of circuit designers and of those who manufacture integrated circuits. Minimization is important from a cost point of view because fewer circuits reduce the cost of manufacture of a complicated logical array. Minimization is also important from the context of reliability because the more components there are the greater the likelihood that one of them may fail.

Although minimization has been a goal, achieving that goal has been met with limited success. Only the simplest of logic problems have been susceptible

to formal simplification techniques. The more complicated logical problems have simply been too difficult to solve.

The more complicated logical problems, called herein multilevel problems, have not been susceptible to any kind of formalized simplification process that would result in a demonstrably correct simplification of the logic to a simpler form. Multilevel logical representations typically involve cross linkage between terms of a logical expression and in the use of brackets and other separators. Until this point in time there has been no generally applicable technique for simplifying multilevel logical circuitry.

SUMMARY OF THE INVENTION

The invention is directed to apparatus, systems, methods, computer program products and other techniques for simplifying logical circuitry and other multilevel minimization problems.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the system of the present invention will be apparent from the following description in which:

Figure 1 shows an exemplary set of unit vectors that will be used in vector representation of logical schema.

Figure 2 shows the vector representation of an exemplary logical schema S.

Figure 3 represents the schema S with a partial minimization achieved by deleting opposed couples on the path of the or a greatest symmetry, in this case for the variable p.

Figure 4 represents Figure 3 with redundancies removed.

Figure 5 shows Figure 4 with the procedures SS, LL and ER described below applied for the variable q.

Figure 6 shows the process described below including SS, LL and ER applied to Figure 5 for the variable r.

Figure 7 shows the process described below including SS, LL and ER applied to Figure 6 for the variable s. Figures 6 and 7 are identical.

Figure 8 shows unminimized S in gate or circuit form.

Figure 9 shows minimized S in gate circuit form.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description relates to the solution to an exemplary as opposed to one multilevel logic minimization problem: find the or a shortest equivalent for the schema S.

S: [-p(qr-s v q-r-s)] v p-q-rs v [p(q(r-s v –rs v –r-s))].

Some comments on notation will facilitate the understanding of the invention and the techniques described herein. In the schema listed above, there are four literals, namely p, q, r and s. When represented in text, the dash before a literal indicates the negation of the literal. In other words, "-s" means "not s." Another representation for "not s" is the literal s with an over-score sometimes called "s bar". Since word

3

processing programs don't easily allow for literals to be over-scored, it is easier to represent the "not" state by using a dash before them when typing. However, in the drawings, the over-score will commonly be used in lieu of the dash. In the schema above, the separator "v" represents a disjunction or an alternation. Terms joined by a "v" are related by an "or" function. Literals and expressions that are adjacent to each other are considered to be conjunctions such that both terms or expressions must be true for the overall expression to be true. Conjunction corresponds generally to a logical "AND" function and logical disjunction or alternation corresponds to logical "OR" function.

The methods and techniques disclosed for simplifying the exemplary schema shown above, will allow one skilled in the art to find solutions of the same sort for all other multilevel minimization or simplification problems.

Figure 1 shows a set of unit vectors that will be used in vector representation of the exemplary logic schema discussed above. There are four literals, namely **p, q, r** and **s**. Each of the literals have their respective negations represented by a vector. For example, the literal **p** is represented by vector in the horizontal plane with an arrowhead pointing toward the origin **O**. The negation of **p** is also represented by a vector on the horizontal plane but pointing in the opposite direction toward the origin **O**. The other literals have similar representations.

When representing the logical schema, described above, these unit vectors will be utilized. In this notation, only four literals are shown, because that is all that is needed for this particular logical schema. However, the set of unit vectors may be increased, as needed depending on the complexity of the problem. In other words, although only four literals are shown, the unit vector approach may be expanded to accommodate any number of literals. Each literal will have a unit vector associated with a direction and the arrowhead of which represents whether the literal or its negation is represented depending on the direction in which the arrowhead points. Disjunctions or alternations are represented by forks in the vector system.

The general multilevel minimization procedure is as follows.

Step one--Represent the multilevel schema or schemata in vector format as shown below.

Step two—Take a or the greatest symmetry (GS) within the system. In the case of the exemplary schema, these would be the expressions offered by p(q)r-s v –r-s)) and (-p)qr-s v q-r-s)), which contain a couple (e.g. **p, - p**). A literal and its negation when represented in vector notation will sometimes be referred to an opposing couple. The **p** couple vectors will be eliminated and the two expressions linked by those will be joined at a common point in this case at the origin. This is sometimes referred to as a symmetry slide or SS. This will become more apparent as we discuss this specific example hereinafter.

Step three—Leave any routes which are not part of the symmetry of the selected opposing elements as described hereinafter.

Step four—Repeat these steps with respect to opposing couples of each of the other literals.

Step five—Translate the final equivalent vector representation of the simplified schema into a standard schema in prepositional logic.

In the example to be described more in detail hereinafter, the original logic implementation of the schema was a circuit with 13 two input and gates, 5 joins (or gates) and 10 inverters

The unsimplified implementation is shown in Figure 9 in circuit form. The simplified form of the schema which results from the techniques described above and to be described in more detail below, is a circuit which has 3 two input and gates, one join (or gate) and two inverters. The latter describes a significantly cheaper, faster and more reliable circuit.

The methods and techniques disclosed for the following example will allow one skilled in the art to find solutions of the same sort for all other multilevel logic minimization or simplification problems.

As noted above, the invention will be exemplified by simplifying the following schema, as an example:

$$S: \text{[-p(qr-s v q-r-s)] v p-q-rs v [p(q(r-s v -rs v -r-s))].}$$

The first step involved in minimization in the schema is to represent the schema in a vector notation. This will be done showing line representations of vectors as in Figure 2 but this can also be done utilizing representation of vectors inside a computer. An origin, O, serves as a reference point for representing the logical schema in vector notation.

Turning first to the area described as term 1 in Figure 2, beginning with the origin, the "- p" literal is represented by a horizontal vector pointing into the origin. The conjunction of q with that "-p" vector is illustrated by a q vector attached to the tail of the "-p" vector. At the end of the q vector there are two vectors r and $-r$. Note that both of the disjunctions contained within the parenthesis of the first term are q such that the vector q is actually traversed twice and then the paths diverge.

The first term inside the parenthesis of the first term, namely $qr-s$, branches over the path labeled r and $-s$, that is vertically above the junction of the q, r and $-r$ vectors. The second portion of the first term also begins with the junction of the q,r and $-r$ vectors and proceeds over the $-r$ vector to the $-s$ vector.

One should note that disjunctions, in this notation, are visually identifiable by forking of vectors off of a junction point. Therefore the disjunction represented by the "v" in the first term of the schema is shown as the separate r and $-r$ paths of the intersection with q. Conjunctions, on the other hand are represented by placing the appropriate vectors head to tail or tail to tail as appropriate. Similarly, terms 2 and 3 of the schema both begin with a p vector. In Figure 2, the p vector is shown with a "2" indication adjacent. This implies that there are two traversals of the p vector path one for each of terms 2 and 3.

5

Going to term two, which is **p-q-rs**, the vector **p** is followed from the origin to the right the vector **–q** along the **–q** vector to the intersection with **–r** followed by the **s** vector.

Note that in Figure 2 that the intersection of the **–q** and **–r** vectors on the **–s** vector is an artifact of the representation and implies no logical relationship. It is merely an incident of the rendition.

Term 3 of the schema also begins by traversing the **p** vector to the right and follows then the **q** vector to the disjunction represented by **r** the junction of **q**, **r** and **–r** vectors in the term three space. The first term within the third term follows the **r** path to the **–s** vector. The second term in the third term follows the **–r** path to the **–s** vector. The third term follows the **–r** vector to the **s** vector. As before, the **–r** vector is shown with a "2" adjacent to show that the vector is traversed twice in the representational notation.

Once a schema is represented in vector notation as shown in Figure 2, one can begin the process of simplification. The simplification process is centered about opposing couples such as vectors **p** and **–p**.

The question of " symmetry" can best be illustrated with reference again to Figure 1. Figure 1 has, in addition to the unit vector representations, a set of compass directions indicated on the drawing. If one were to describe the route to be taken from a physical location on the face of the earth to a destination, one could describe it as a set of directions to be followed. For example one could say: "Drive north one mile then drive east one mile then drive southeast a half a mile and then south for a quarter of a mile to the destination." This is analogous to a logical representation where a portion of the same route is traversed, coming from different beginning points.

Referring again to Figure 2, one can see that there is similarity in vector representation S between the term that follows the **–p** vector and those that follow the **p** vectors. For example, at the end of each **p** and **–p** vector there is a vector **q** which then intersects with vectors **r** and **–r**, the **r** vector of which intersects with a **–s** and the **–r** vector of which intersects with the **–s**. Thus, there is great similarity between the paths followed once the **p** and **–p** vectors have been traversed. It is exploiting the symmetry that permits a reduction in the complexity of the logical representations.

Returning again to Figure 3, we focus on the **p, -p** opposing couple. Our intention is to remove one instance of the **p** vector and counter balance it against the **–p** vector in the simplification process. Accordingly, the portion of the vector representation to the right of the **p** vector that corresponds to

6

the representation to the right of the –p vector will be superimposed; that is the corresponding vector will slide (symmetry slide) over so that they coincide at a single point. In this case, since the –p vector has been "canceled" by one instance of the **p** vector, the vector branches off of the –p vector will slide so that the **q** vector points to the origin and similarly, the corresponding symmetrical vectors to the right of the **p** vector will be slid to the origin so as to coincide with those that come from the removal of the –p vector. The result of the symmetry slide is shown in Figure 4. Note that the portion of the third term that corresponds to the disjunction –rs does not have a corresponding counterpart in the symmetrical portion of term 1. Accordingly, one instance of the **q** vector one instance of the –**r** vector and the **s** vector do not slide and remain where they are. That is what is meant by the notation above Figure 4 about keeping multiple paths open.

In Figure 4, the **r** and the –**r** vectors are shown in dashed form to indicate that they are the next opposing couple that will be considered for simplification. As shown in Figure 4, both the **r** vector and the –**r** vector intersect a –**s** vector. The –**s** vector represents a symmetrical feature that can be simplified. Accordingly, the **r** and the –**r** vector are eliminated and the –**s** vector from both ends slide (symmetry slide) onto the end of the **q** vector with results in the representation shown in Figure 6.

Figure 7 shows that we will next focus on the **q**, -**q** opposing couple. Again, symmetry is shown in that each of the **q** and –**q** vectors terminate in a –**r** vector connected to an **s** vector. Again the **q** and –**q** vectors are eliminated and the –**r** and **s** vectors that are attached to the endpoints thereof slide together to the common point at the end of the **p** vector. The resulting simplified logical expression is shown in Figure 8. The simplified schema representation, S*, is as follows.

$$S* = q\text{-}s \ v \ p\text{-}rs.$$

When comparing the original schema S with S*, the degree of simplification is significant.

Figure 9 shows a logical circuit which implements the schema S and Figure 10 shows a logical circuit which implements the simplified schema S*. The degree of simplification is nicely apparent when comparing the two circuits. Implementing schema S as shown requires 13 two input gates 5 or gates and 10 inverters. Figure 8, on the other hand has 3 two input gates, 1 or gates and 2 inverters. The circuit of Figure 10 is vastly cheaper, faster and more reliable. It is faster because there are fewer logic decisions that need to be made resulting in a faster response time.

To demonstrate that the logic of the simplified schema S* is logically equivalent to the logic of the original schema S, Appendix A sets forth a truth table constituting a proof that the two circuits are logically identical although one is vastly cheaper, faster and more reliable that the other.

Accordingly, applicants have demonstrated techniques that are well suited for simplification of logic circuitries particularly those of the multilevel nature. Applicants have demonstrated techniques for minimizing logic circuitry which are suitable for implementation in CAD CAM programs for the design in implementation of logical circuits.

The techniques described herein can be implemented routinely in software operating in such devices or in general purpose computers. The use of these techniques will permit vast simplification in the production design in manufacture of logical circuits, including integrated circuits

Although the invention has been described herein with respect to a specific example, numerous extensions and applications of the techniques described herein will be apparent to one skilled in the art. The invention is not limited to the description herein but rather is reflected in the claims below.